Claims 1 to 11 and 28

The Examiner concedes in paragraph 4 that Arvind *et al.* has not specifically taught "Distributing at least one instruction for data processing to one data processing unit, before the data processing unit is available to process the instruction," but then combines Hennessy with Arvind *et al.* to hold that the claimed invention is obvious. The Examiner's position is that reservation stations taught by Hennessy are well known in the art to buffer instructions waiting to execute and buffer required operands and data as they become available (Hennessy, pages 252 - 253), for the desirable purpose of executing the instructions as soon as all of the operands required for execution become available, and that this eliminates the need to retrieve the instructions and operands from registers at the time of execution which speeds up instruction execution time.

However, the applicant submits that these two approaches are described in the context of two completely different processor architectures, and a person of ordinary skill in the art would not be motivated to combine these references.

Arvind *et al.* teach a purely data-driven processor architecture where an instruction becomes available only after a related data packet is received by the data processing unit for processing (because the instruction address is attached to the data packet as a token and the processor retrieves the instruction from memory after it receives the data packet with the corresponding token). Hennessy teaches a completely different approach to processing – a conventional instruction-driven processor architecture, where instructions are made available for distributing to data processing unit before the data processing unit is available to process the instruction. Hennessy's instruction-driven architecture is a two-step process: 1) Processing (scheduling) the instruction and sending for data; and 2) Processing (executing) the data and sending out results.

With respect, it would not be obvious to one of ordinary skill in the art to combine these references in the manner suggested by the Examiner, nor is it a simple matter of adding Hennessey's 'reservation stations' to arrive at the invention.

One cannot merely combine data-driven and instruction-driven approaches. It is not possible in a purely data-driven processor architecture such as Arvind *et al.* to distribute the instruction to a data processing unit before the data processing unit is available to process the instruction the way Hennessey does, since the processor does not at this stage know what instruction it must process.

The present invention combines these two approaches, mixing instruction-driven and data-driven processor architectures so that instructions are distributed as in conventional instruction-driven processor architecture as taught by Hennessy, but the data is processed as in data-driven processor architecture as taught by Arvind *et al.* To accomplish this, the present invention maintains records of processed instructions with outstanding data requests. This allows for instructions to be fetched while previous instructions are awaiting data for processing, but also provides other advantages, for example, it allows for the processing of instructions by a single processor.

Thus, while Hennessey (as an instruction-driven architecture) requires multiple instances of hardware to process multiple instructions simultaneously, because the present invention only keeps records of processed instructions with outstanding data requests, the processing of instructions in the system of the invention is not hardware-dependent and can. This not only saves on hardware, but also allows one to merely increase size of the outstanding instructions record in order to accommodate a larger latency (whereas Hennessy would have to add more instances of hardware in order to process more instructions simultaneously).

The applicant respectfully submits that the combination of steps as presently claimed, including the steps of distributing at least one instruction for data processing to one data processing unit of the multiple data processing units *before the data processing unit is available to process the instruction* – which the Examiner agrees is not taught by Arvind *et al.* – storing a record of the requested data packet, and associating with each data packet sent out a data token showing the readiness of the packet for further processing, is not obvious in view of the cited references.

Claims 18 to 27

The Examiner agrees that Arvind *et al.* does not teach that the data path is separate from the instruction path, but asserts that making something separate is not a patentable difference. With respect, the Examiner is applying a case that dealt with a different invention too dogmatically. Each case is specific to its facts, and it cannot be said as an absolute rule that making something separate is not a patentable difference. There are many situations where making something separate can be a patentable difference; the question is always whether the modification is unobvious. In the present case, by separating the instruction path from the data path, the inventors break up one pipeline inside the processor into two pipelines – an instruction pipeline and a data pipeline – which avoids processor stalls.

Processor stalls can be separated into three categories: stalls inside the instruction pipeline, stalls between the instruction pipeline and the data pipeline, and stalls inside the data pipeline. By separating the instruction path from the data path, stalls inside the instruction pipeline are avoided because instructions can be distributed as in conventional instruction-driven processor architecture; and stalls inside the data pipeline are avoided since data packets are processed as in a data-driven processor architecture. This makes the combination of instruction-driven and data-driven processor architectures used in the present invention more efficient.

But this is not obvious based on the cited art, because its advantage arises in the hybrid system unique to the present invention, which mixes instruction-driven and data-driven processor architectures. A person of ordinary skill in the art would not have been motivated to separate the instruction path from the data path in a purely instruction-driven processor architecture as taught by Hennessy or in a purely data-driven processor architecture as taught by Arvind *et al.*, nor would there be an advantage to doing so.

The applicant therefore respectfully submits that the present claims recite an invention that is novel and unobvious and are allowable. Favourable reconsideration and allowance of this application are respectfully requested.

A Petition for an Extension of Time requesting an extension of one month for filing the subject response is enclosed. The Commissioner is authorized to charge any deficiency or credit any overpayment in the fees for same to our Deposit Account No. 500663. A signed copy of this page is enclosed if required for this purpose.

Executed at Toronto, Ontario, Canada, on April 24, 2006.

DANIEL GUDMUNSON, ALEXEI KROUGLOV, ROBERT COLEMAN

Mark B. Eisen
Registration No. 33,088
(416) 971-7202, Ext. 242
Customer Number: 38735

MBE:lf

Processor stalls can be separated into three categories: stalls inside the instruction pipeline, stalls between the instruction pipeline and the data pipeline, and stalls inside the data pipeline. By separating the instruction path from the data path, stalls inside the instruction pipeline are avoided because instructions can be distributed as in conventional instruction-driven processor architecture; and stalls inside the data pipeline are avoided since data packets are processed as in a data-driven processor architecture. This makes the combination of instruction-driven and data-driven processor architectures used in the present invention more efficient.

But this is not obvious based on the cited art, because its advantage arises in the hybrid system unique to the present invention, which mixes instruction-driven and data-driven processor architectures. A person of ordinary skill in the art would not have been motivated to separate the instruction path from the data path in a purely instruction-driven processor architecture as taught by Hennessy or in a purely data-driven processor architecture as taught by Arvind *et al.*, nor would there be an advantage to doing so.

The applicant therefore respectfully submits that the present claims recite an invention that is novel and unobvious and are allowable. Favourable reconsideration and allowance of this application are respectfully requested.

A Petition for an Extension of Time requesting an extension of one month for filing the subject response is enclosed. The Commissioner is authorized to charge any deficiency or credit any overpayment in the fees for same to our Deposit Account No. 500663. A signed copy of this page is enclosed if required for this purpose.

Executed at Toronto, Ontario, Canada, on April 24, 2006.

DANIEL GUDMUNSON, ALEXEI KROUGLOV, ROBERT COLEMAN

Mark B. Eisen
Registration No. 33,088
(416) 971-7202, Ext. 242
Customer Number: 38735

MBE:lf